AD-A057 465 STANFORD UNIV CALIF DIGITAL SYSTEMS LAB F/G 9/2
SPRINT - AN INTERACTIVE SYSTEM FOR PRINTED CIRCUIT BOARD DESIGN--ETC(U)
NOV 77 W M VANCLEEMPUT, T C BENNETT, J A HUPP N00014-75-C-0601
UNCLASSIFIED DSL-TN-131 NL

1 OF 1
ADA
057465

END
DATE
FILMED
9-78
DDC

LEVEL II

12

SPRINT - AN INTERACTIVE SYSTEM FOR PRINTED CIRCUIT BOARD DESIGN: FUNCTIONAL CHARACTERISTICS.

by

W. M. vanCleemput, T. C. Bennett, J. A. Hupp, K. R. Stevens

Technical Note No. 131

DSL-TN-131

November 1977

28p.

Digital Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305

This work was supported by the U. S. Energy Research and Development Administration under Contract E4-76-C-03-0515 and by the Joint Services Electronics Program under Contract N-00014-75-C-0601.

SPRINT - AN INTERACTIVE SYSTEM FOR PRINTED CIRCUIT BOARD DESIGN: FUNCTIONAL CHARACTERISTICS

W. M. vanCleemput, T. C. Bennett, J. A. Hupp, K. R. Stevens

Technical Note No. 131

November 1977

Digital Systems Laboratory
Stanford University
Stanford, California 94305

ABSTRACT

The SPRINT system, designed and implemented on the IBM 370 system at the Stanford Linear Accelerator Center (SLAC), is a highly interactive system for the layout of printed circuit boards.

At the choice of the designer, a layout can be generated manually, automatically or usually by a combination of man-machine interaction and automatic design algorithms.

INDEX TERMS: Computer-aided design, printed circuit boards, structured design, design automation.

Also published as Computation Group Technical Memo No. 192, Stanford Linear Accelerator Center, Box 4349, Stanford, California 94305.

1. Introduction

The objective of the SPRINT system is to allow interactive computer-assisted design of printed circuit boards. The SPRINT system allows for the manual placement of critical components and for automatic placement of other components. The interconnection routing module allows for manual routing of critical connections and for automatic routing of non-critical connections. The current system is limited to two signal layers, but a future expansion to multi-layer boards has been planned.

The input to the system is in the form of an input language called the Structural Design Language (SDL). In the future the SDL description will also be used as the input to a logic simulator, to a fault test generation/simulation system and to an automatic logic diagram generation capability.

The current system is implemented in MORTRAN and FORTRAN IV on the IBM 370 system at SLAC and makes use of a Tektronix 4013 terminal. The SDL compiler is implemented in SPITBOL, a SNOBOL dialect. Currently, the output of the system is in the form of a Calcomp plot, from which the artwork must be generated manually. Facilities for direct generation of final artwork are being developed.

The SPRINT system is a part of an integrated design automation system, which is currently being developed at SLAC. This system will allow a designer to specify his circuit as a logic diagram using an interactive graphics terminal, to verify his design using a logic simulator, to design a wire-wrap or printed circuit board, to layout an integrated circuit chip, and to generate test patterns. The basis of this system is SDL [1,2], a hierarchical language for describing structural properties of digital systems at various levels of detail.

Currently, the SDL system, the SPRINT system for PC design and an interface to the TESTAID [3,5] logic simulator have been completed. The other parts of the system are in varying levels of completion, and are discussed in section 7.

2. System Overview

The current system consists of the following major programs:

1) SDLCOMP: The SDL compiler translates the circuit description provided by the user into an internal format. It makes use of a logical library. This logical library contains information for each component type used.

2) SDLPCGEN: This program takes the sequential output file generated by the compiler, and combines it with information retrieved from the physical library and the board description file. The output is a sequential file containing all the information necessary to initialize the design file.

3) LOADFILE: This program initializes the direct-access design file using the output generated by SDLPCGEN.

4) PLACER: This is an interactive subsystem that allows the user to place components on the board.

5) MWIRE: interactive subsystem for routing critical and multiple-width wires manually.

6) HIWIRE: batch router for automatic routing of non-critical connections.

7) VIAELIM: batch program for elimination of unnecessary via holes from the design.

8) PCPLOT: a program for generating plots and artwork for a completed design.

The overall structure of the system is illustrated in Figure 1.

3. Input to the System

Currently, a design must be encoded from a logic diagram into a machine-readable form. The language used for this description is SDL (Structural Design Language). The description of a circuit in SDL can also be used for logic simulation, fault test generation, circuit analysis and IC layout.

The SDL language allows format-free entry of a circuit. It performs checking of the input where possible. Common errors such as two signals connected to the same pin will be detected quite easily. It has a powerful hardware macro description capability, which encourages the use of a structured design methodology.

Figure 2 shows the encoding in a subset of SDL for a simple example. The description starts with an identification section where user name, circuit name, level and purpose of the description are specified.

This is followed by a declaration of all types to be used in this circuit and by a declaration of an external connector called CONN20A. The signal nets are then specified by a number of component-name, pin-name pairs. In order to enchance the error detecting capability,

the user may enter the signals in their dual form after the CROSSCHECK statement. Here, for every component, a list of net-name pin-name pairs is given.

The example only illustrates some of the more primitive constructs in SDL. It is, for example, possible to declare and use macros to specify busses (rather than simple signals), to specify inputs and outputs for components, to specify equivalence between certain parts of a component (e.g. 4 equivalent NAND gates in a 7400 package).

These more advanced primitives are described in detail in [1,2].

The system makes use of two libraries:

1) a logical library which contains information about the logical characteristics of each component, e.g. number of pins, pin names, and which pins are inputs, outputs, power and ground.

2) a physical library which contains physical characteristics for every component, e.g. size, pin locations, and obstructions.

In addition to the system's logical and physical libraries, which describe most 7400-series IC's and common discrete components, the user may specify private libraries. Their descriptions add to or supercede those in the system libraries to handle special user requirements.

Finally, it is necessary to physically describe the printed circuit board to be used. An example of such a free-format description is given in Figure 3.

4. The Placement Subsystem

In order to allow maximal flexibility in placing components, this subsystem is highly interactive. Depending on the nature of the design and the inclinations of the designer, placement may be done completely manually or completely automatically. Usually the placement process consists of a combination of both manual placement and algorithmic placement improvement. The SPRINT system reflects this philosophy in the following four major steps:

1) Placement of critical components. This step is manual and allows the designer to carefully place components for which he considers placement to be critical. The so-called "critical" components will not be moved by the automatic placement optimization algorithms although the designer can do so at all times. The remaining components will be classified as "automatic" or "discrete" components. In the context of this system, discrete components are resistors and capacitors for which automatic placement is difficult. An example of this is decoupling capacitors between power and ground in TTL logic or termination resistors in ECL.

2) Initial placement of the "automatic" components. In this phase the designer has the choice between two different placement methods:

> manual initial placement: useful if the designer has a good feeling for how the placement should look.
>
> random initial placement: random placement of components in available locations. This can be useful for evaluating several runs of the placement improvement algorithms for different starting points.

Implementation of one or more constructive placement algorithms, e.g. [4], for producing suboptimal initial placement is planned.

3) Placement improvement: Currently, only pairwise interchange is available for optimizing placement, with the possibility of later adding force-directed placement and other techniques. The designer does have the option of selecting either ECL (minimum chains starting at an output or external pin) or TTL (minimum enclosing rectangle) rules for wirelength calculations made to evaluate interchanges. Actual pin coordinates are used for all calculations.

4) Placement of "discrete" components: In this last phase, the designer manually places all remaining components.

Movement back and forth between these placement phases is quite easy, giving the designer a great deal of freedom in how he approaches the design. One interesting approach that has been experimented with is that of hierarchical placement. Sets of similar-size components are sequentially chosen as the current automatic components; the board is divided into appropriate-sized cells, and the set is automatically placed and optimized. In this way a near-optimal placement of most components can be obtained fairly easily even though they may be of a wide range of sizes.

5. The Conductor Routing Subsystem

5.1 Introduction

This subsystem consists of three major parts:

1) MWIRE: an interactive system for connecting critical wires.

2) HIWIRE: a batch program for automatic routing of the interconnections with vertical segments on one layer and horizontal segments on the other side.

3) VIAELIM: a batch program for the elimination of unnecessary vias from the routed design.

5.2 Manual prewiring of critical connections (MWIRE)

The MWIRE program allows a user to route critical connections manually, using an interactive graphics terminal. It can also be used for routing multi-width wires (e.g. for power and ground connections). Only valid connections can be made; i.e. the system checks a number of design rules. The system will detect intersections of signals and incomplete routing of signals. Since automatic routers become rather inefficient when varying-width connections are allowed, it was decided to route this type of signal manually. The manual router allows for both vertical and horizontal line segments on the same board layer.

5.3 Batch Routing Program (HIWIRE)

HIWIRE is an automatic batch router using a modified version of Hightower's algorithm [7,8] to route two-sided printed circuit boards. The algorithm provides a relatively high connection-completion rate using a relatively small amount of computer time. It is, however, only a heuristic and will not necessarily find the optimal path betwen two pins or even find a path at all when one might exist. A fair amount of optimization is done, though, to maximize the chance of finding a good path.

HIWIRE uses the design file for input and output. PLACER must have been run on the design file to provide pin locations for the networks, and MWIRE may have been run to route multi-width busses or critical wires and to insert wiring obstructions. After routing is completed (as far as possible), the design file is updated by inserting information about the segments and vias. This information can then be used by VIAELIM to minimize the number of vias or be plotted. In addition to the design file output, HIWIRE can produce lists of all connections it has made for each net, the pins it has been unable to connect, and the paths it tried.

As stated above, HIWIRE can only handle boards with two signals planes at the present time. Routing is done with all horizontal segments on one plane and all vertical segments on the other, though VIAELIM will try to bring as much of each path to one layer of the board as possible. The board size, number of pins, number of nets, etc., which can be handled is fairly arbitrary; most parameters are macros which can be easily changed at compile time. Another factor in applicability is the quality of the routing obtained for a given circuit. All that can be said here is that HIWIRE will do the best it can with the placement it is given. Care must be taken to avoid producing overly congested areas on the board (e.g. near an external connector) or

manually routing too many (or too long) 'wrong' side wires (which act as obstructions to HIWIRE). The ease with which a placement can be modified, combined with the relatively fast execution of HIWIRE provides a system for adaptively improving the router's performance on a given board. Hopefully, all connections can be completed or few enough will be left that only minor editting by hand will be necessary.

5.4 VIA ELIMINATION PROGRAM (VIAELIM)

VIAELIM is a batch program designed to eliminate unnecessary via holes from the automatic routing produced by HIWIRE. This is a fairly important function, since HIWIRE routes all paths with horizontal and vertical segments on different board layers, resulting in a large number of vias. If carried through to the final board, these vias would increase the board's cost and decrease its reliability.

Another property of VIAELIM is that, in changing the locations of segments from one side of the board to the other to eliminate vias, it may clear previously blocked paths and allow further routing to be done by another pass through HIWIRE. Segments now on the 'wrong' side will act as new obstructions to automatic routing, so one should not expect large numbers of previously unroutable pins to be routed by this process. Some improvement in the completion ratio, however, could occur.

VIAELIM is partly based on an algorithm presented by Hashimoto and Stevens [6]. The modifications to this algorithm are discussed in [8]. Basically, it models the routing as a graph in which the nodes represent disjoint sets of mutually intersecting segments and the edges represent vias where segments from different groups connect. Each node is assigned one of two colors (indicating that segments in the set either stay where they are or switch sides) attempting to minimize the number of edges between two nodes of the same color. Edges between nodes of

different colors represent vias that can be eliminated. This procedure is equivalent to trying to find the maximal bipartite subgraph of the modelling graph. Routing information is obtained from the design file and is updated after via elimination is completed.

6. An Example

Figure 4 shows a checkplot, drawn on a matrix-type plotter of the completed design of a simple circuit. The SDL encoding of this circuit is given in Figure 2. The time required to encode the circuit diagram and to correct it after running it through the SDL compiler was about two hours. To do the interactive placement required less than fifteen minutes. The CPU time for doing the automatic routing was 1.9 seconds for 100% completion. The via elimination program reduced the number of vias from 103 to 50, a significant improvement. The CPU time for doing this via reduction was 1.2 seconds (IBM 370/168).

Figure 5 shows the artwork derived from the computer-generated layout for another sample design.

7. Results and Conclusions

Table 1 shows the summary of some boards that were designed using the system. Several medium to large boards were designed using the system. The results for some of these boards were compared to other existing printed circuit board design systems and were, for the most part, better.

To illustrate, example 2 required only a relatively small amount of human designer time to arrive at an optimal placement, starting from a carefully chosen manual initial placement. The amount of CPU time spent in the routing and via elimination phases for this board is rather small (about two minutes).

Because of the low cost of using the system, a designer may try several options without spending an excessive amount of CPU time as a penalty.

So far, about twenty different boards have been designed successfully. The amount of time required to develop this system was remarkably small: about six man-months for the SDL system, ten man-months for the router/via elimination system, and ten man-months for the interactive placement system.

8. Future Enhancements

Several improvements to the current system are in the design and/or implementation phase:

1) Although SDL input is quite acceptable to many designers, the usefulness of the system can be increased substantially by letting the designer input his design in the form of schematics. Such a system, using interactive storage tube graphics, is currently being developed.

2) Since the physical implementation of prototype systems is often done using wire-wrapping techniques, the expansion of the current system to include automated wire-wrap board design is being considered.

3) Another extension is to include multilayer boards in the routing phase. (The placer is already capable of handling this).

4) Currently a facility is being implemented for automatically generating logic diagrams for each printed circuit board. This subsystem is based on the work described in [9].

REFERENCES

[1] vanCleemput, W. M., "A Structural Design Language for Computer Aided Design of Digital Systems," Stanford University, Digital Systems Laboratory, Technical Report No. 136, April 1977.

[2] vanCleemput, W. M., "A Hierarchical Language for the Structural Description of Digital Systems," Proc. 14th Design Automation Conference, New Orleans, La, June 1977, pp. 377-385.

[3] McClure, R. M., "Fault Simulation of Digital Logic Utilizing a Small Host Machine," Proc. 9th Design Automation Workshop, Dallas, Texas, June 1972, pp. 104-110.

[4] Hanan, M., P. K. Wolff and B. J. Agule, "A Study of Placement Techniques," Journal of Design Automation & Fault Tolerant Computing, Vol. 1, no. 1, October 1976, pp. 28-61.

[5] Hewlett-Packard Co., Testaid III, Programming and Operating Manual, Publication No. 91075093008, November 1976.

[6] Hashimoto, A. and J. Stevens, "Wire Routing by Optimizing Channel Assignment within Large Apertures," Proc. 8th Design Automation Workshop, Atlantic City, NJ, June 1971, pp. 155-169.

[7] Hightower, D. W., "A Solution to Line-Routing Problems on the Continuous Plane," Proc. 6th Design Automation Workshop, Miami Beach, Fla, June 1969, pp. 1-24.

[8] Stevens, K. R., "Conductor Routing and Via Elimination in the SPRINT System," Stanford University, Technical Report, to be published.

[9] Smith, J. A., "Automated Generation of Logic Diagrams," University of Waterloo, Ontario, Canada, Ph.D. Thesis, 1975.

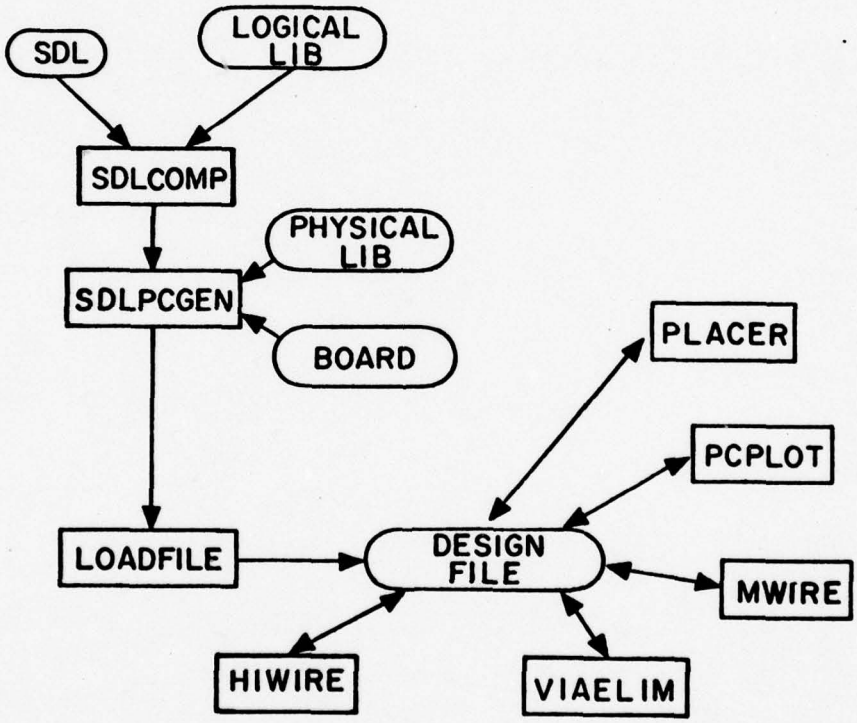

Figure 1: SPRINT System Structure

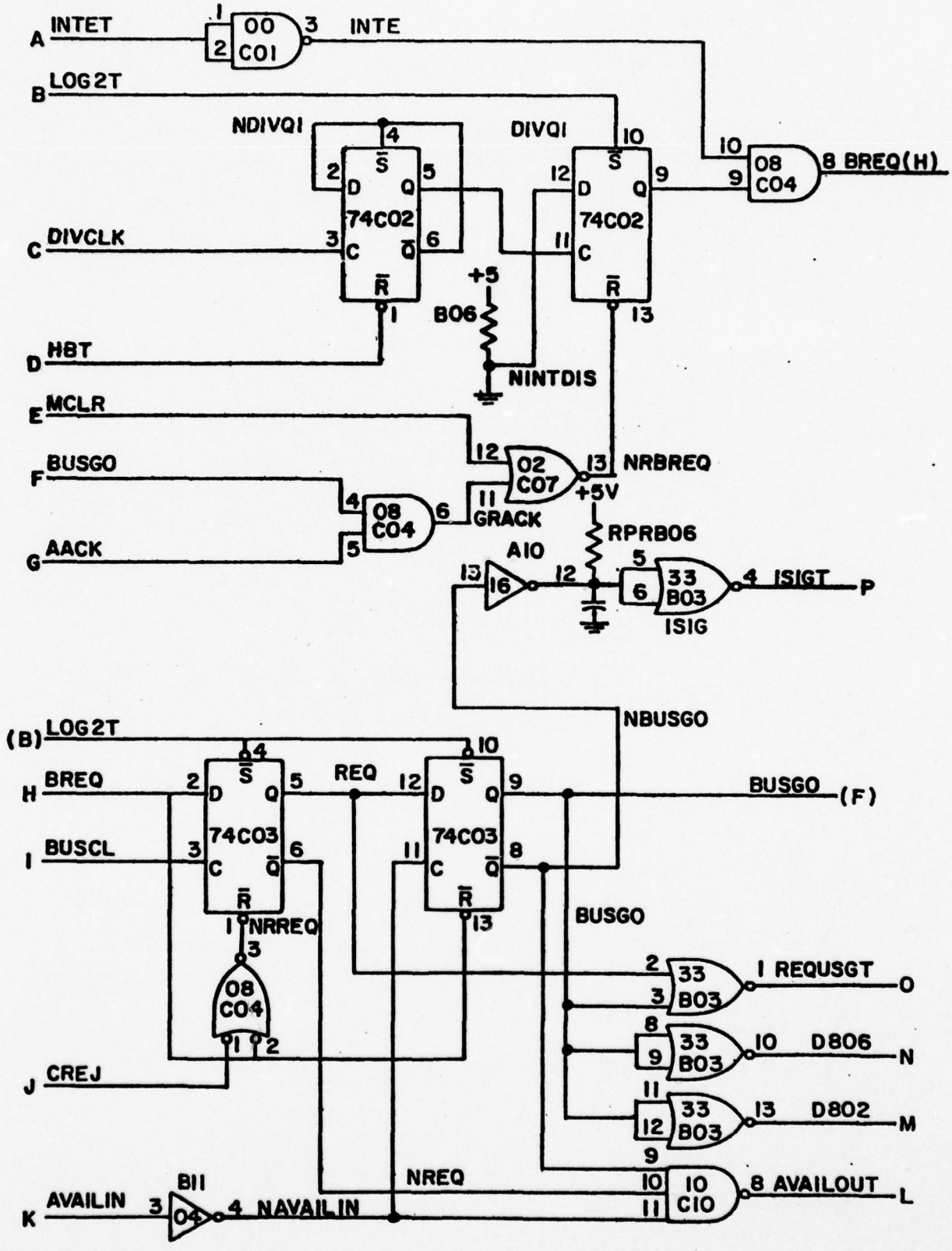

Figure 2(a): Example Circuit

```
USER: WMVC;
NAME:EXA"PLE1;
PURPOSE:PCBGEN;
LEVEL:COMP;
TYPES:7400,7402,7408,7433,7410,7474,7404,7416,RES,CAP;
EXT:CONN20A:A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T;
7400:C01;
7474:C02,C03;
7408:C04;
7402:C07;
7416:A10;
7433:B03;
7410:C11;
7404:B11;
RES:R1,R2;
CAP:C1;
INTET=CONN20A.A,C01.1,C01.2;
INTE=C01.3,C04.10;
LOG2T=CONN20A.B,C02.10;
DIVCLK=CONN20A.C,C02.3;
HBT=CONN20A.D,C02.1;
MCLR=CONN20A.E,C07.12;
BUSGO=CONN20A.F,C04.4;
AACK=CONN20A.G,C04.5;
LOG2T=C03.4,C03.10;
BREQ=CONN20A.H,C03.2,C03.13;
BUSCL=CONN20A.I,C03.3;
CREJ=CONN20A.J,C04.1;
AVAILIN=CONN20A.K,B11.3;
C02Q=C02.9,C04.9;
C02QB=C02.6,C02.2,C02.4;
DIVQ1=C02.5,C02.11;
NINTDIS=C07.13,C02.13;
GRACK=C04.6,C07.11;
REQ=C03.5,C03.12,B03.2;
BUSGO=C03.9,B03.3,B03.8,B03.9,B03.11,B03.12;
NX=C03.8,C11.9,A10.13;
NREQ=C03.6,C11.10;
NAVAILIN=B11.4,C03.11,C11.11;
BREQ=C04.8;
ISIGT=B03.4,CONN20A.P;
ISIG=A10.12,R2.2,B03.5,B03.6;
ISIG=C1.1;
REQUSGT=B03.1,CONN20A.O;
D806=B03.10,CONN20A.N;
```

Figure 2(b): SDL Encoding for the Circuit of Fig. 2

```
D802=B03.13,CONN20A.M;
AVAILOUT=C11.8,CONN20A.L;
PLUS5=R1.1,R2.1;
GND=C1.2,R1.2,C02.12;
END;
CROSSCHECK;
C01:INTET.1,INTET.2,INTE.3;
C02:HBT.1,NINTDIS.13,C02QB.2,DIVCLK.3,C02QB.4,DIVQ1.5,C02Q.9,
    C02QB.6,LOG2T.10,GND.12,DIVQ1.11;
CONN20A:INTET.A,LOG2T.B,DIVCLK.C,HBT.D,MCLR.E,BUSGO.F,AACK.G,
        BREQ.H,BUSCL.I,CREJ.J,AVAILIN.K,AVAILOUT.L,D802.M,
        D806.N,REQUSGT.O,ISIGT.P;
C03:BREQ.2,BUSCL.3,LOG2T.4,REQ.5,NREQ.6,NX.8,BUSGO.9,LOG2T.10,
    NAVAILIN.11,REQ.12,BREQ.13;
C04:BUSGO.4,CREJ.1,AACK.5,GRACK.6,BREQ.8,C02Q.9,INTE.10;
C07:GRACK.11,MCLR.12,NINTDIS.13;
A10:ISIG.12,NX.13;
B03:REQUSGT.1,REQ.2,BUSGO.3,ISIGT.4,ISIG.5,ISIG.6,BUSGO.8,
    BUSGO.9,D806.10,BUSGO.11,BUSGO.12,D802.13;
C11:AVAILOUT.8,NX.9,NREQ.10,NAVAILIN.11;
B11:NAVAILIN.4,AVAILIN.3;
R1:PLUS5.1,GND.2;
R2:PLUS5.1,ISIG.2;
C1:ISIG.1,GND.2;
END;
CEND;
```

Figure 2(b): continued

```
NAME:CAMAC1;
PHYSICAL:0.0,11990.0,11990.4390,10890.4390,10890.5340,
       11990.5340,11990.7230,0.7230,0.0;
LOGSIZE:10400.6300;
LOGOFFSET:500.500;
NCONNECTORS:1;
NAME:CONN1;
NPINS:42;
OFFSET:0.000,0.100,0.200,0.300,0.400,0.500,
       0.600,0.700,0.800,0.900,0.1000,0.1100,
       0.1200,0.1300,0.1400,0.1500,0.1600,0.1700,
       0.1800,0.1900,0.2000,0.2100,0.2200,0.2300,
       0.2400,0.2500,0.2600,0.2700,0.2800,0.2900,
       0.3000,0.3100,0.3200,0.3300,0.3400,0.3500,
        0.3600,0.3700,0.3800,0.3900,0.4000,0.4100;
ORIENTATION:0;
LOCATION:10900.600;
END;
```

Figure 3: Board Description

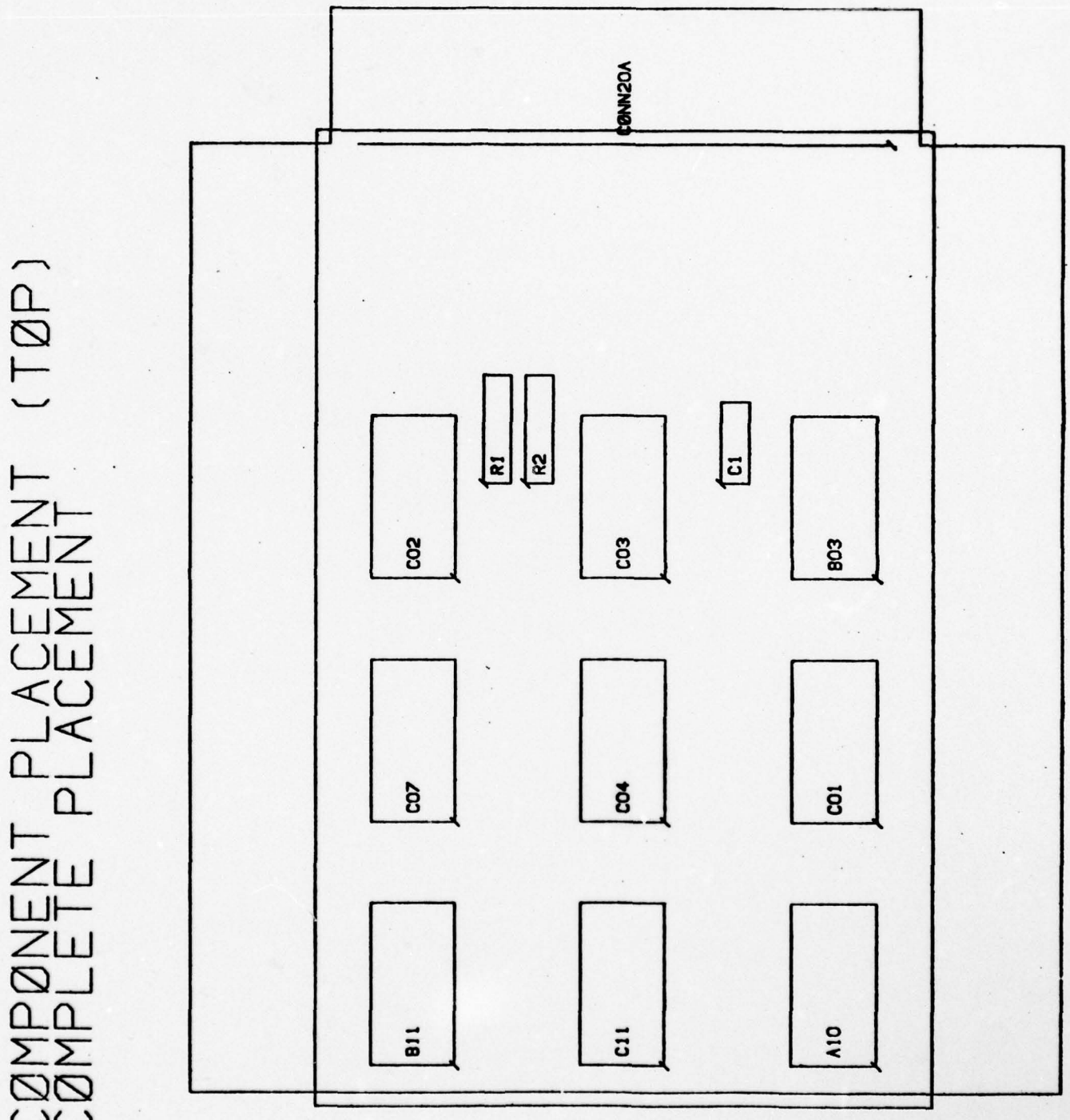

Figure 4(a): Placement Checkplot for the Circuit of Fig. 2

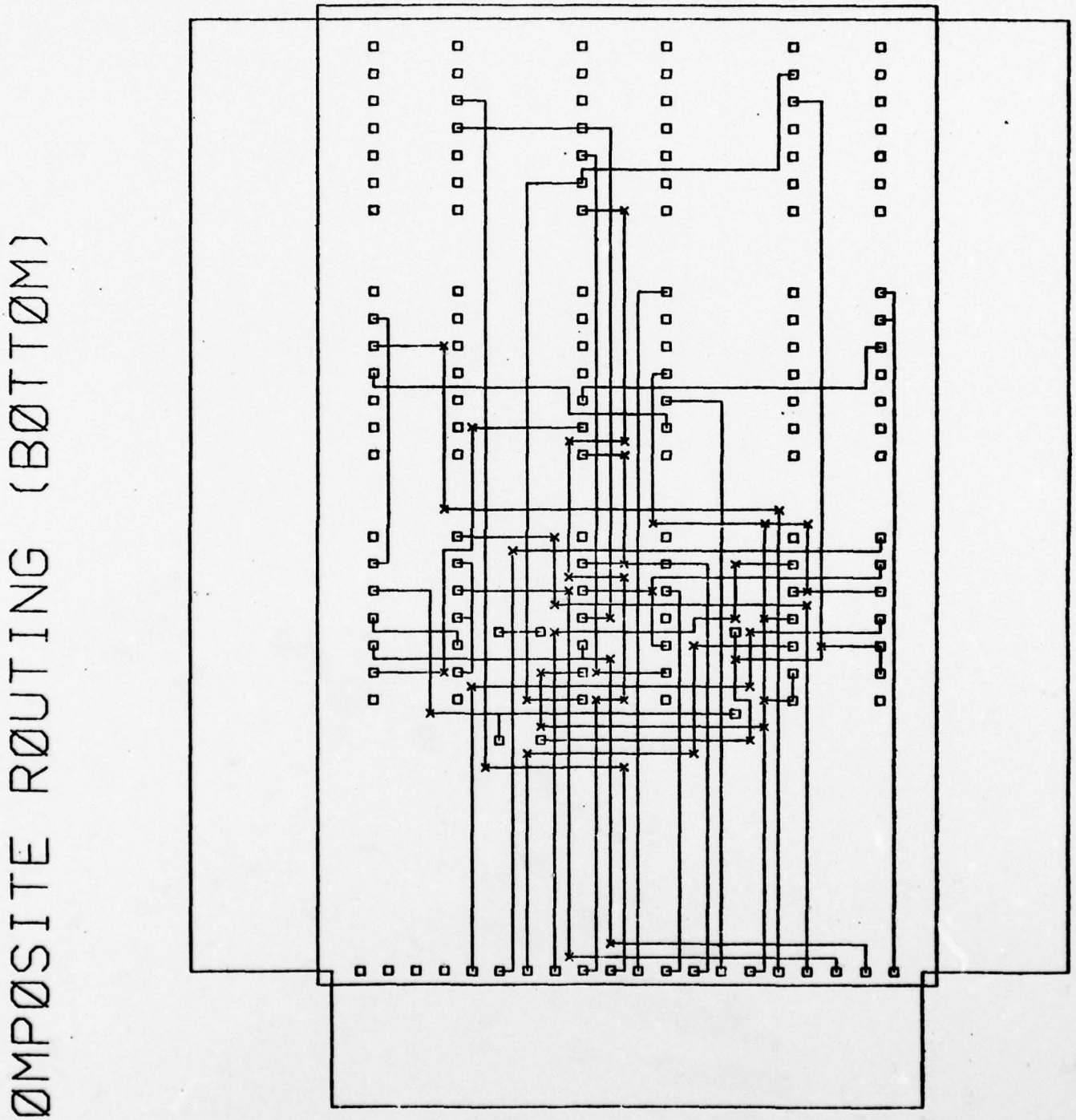

Figure 4(b): Routing Checkplot for the circuit of Fig. 2

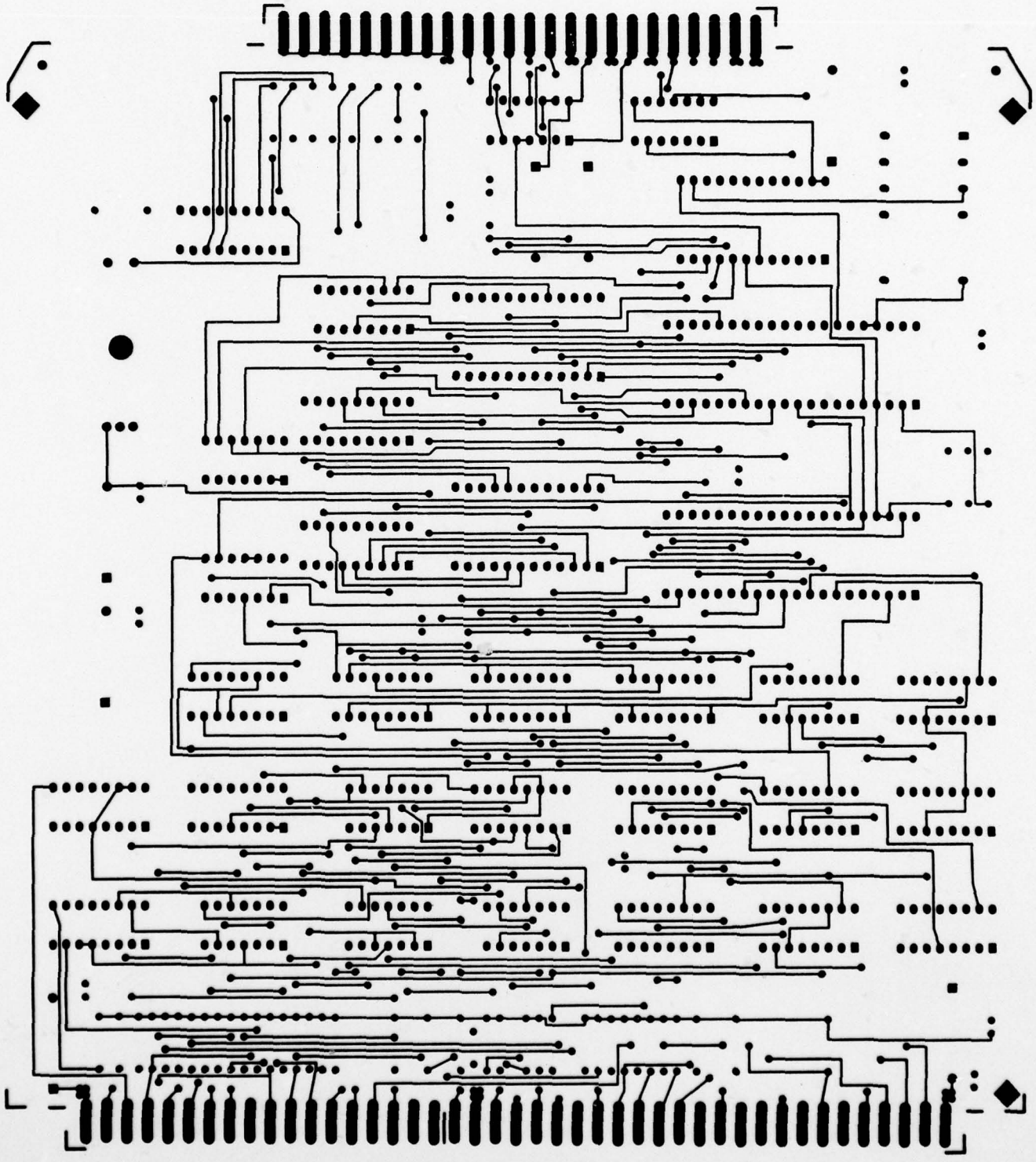

Figure 5(a): Sample Artwork, Top Layer

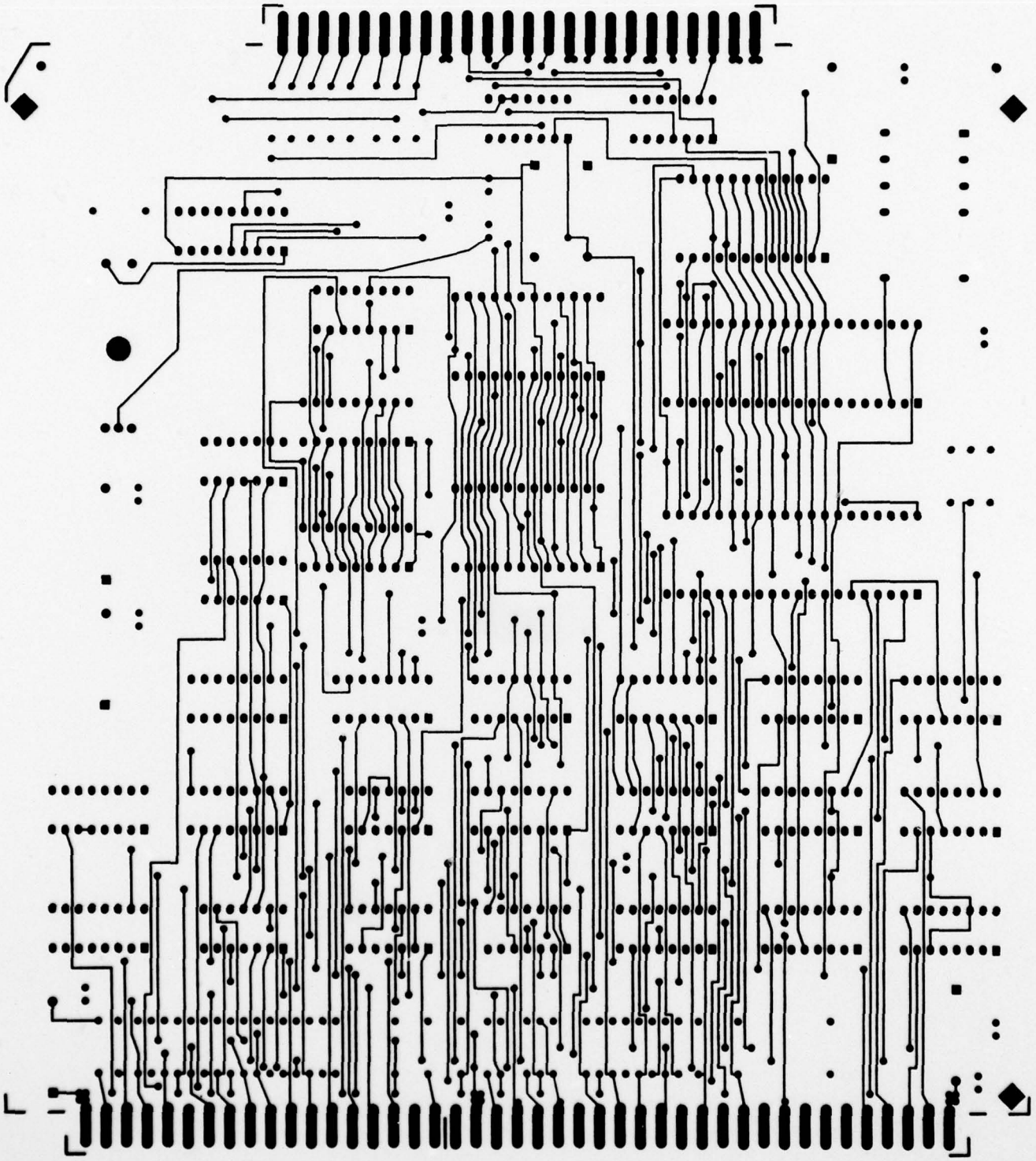

Figure 5(b): Sample Artwork, Bottom Layer

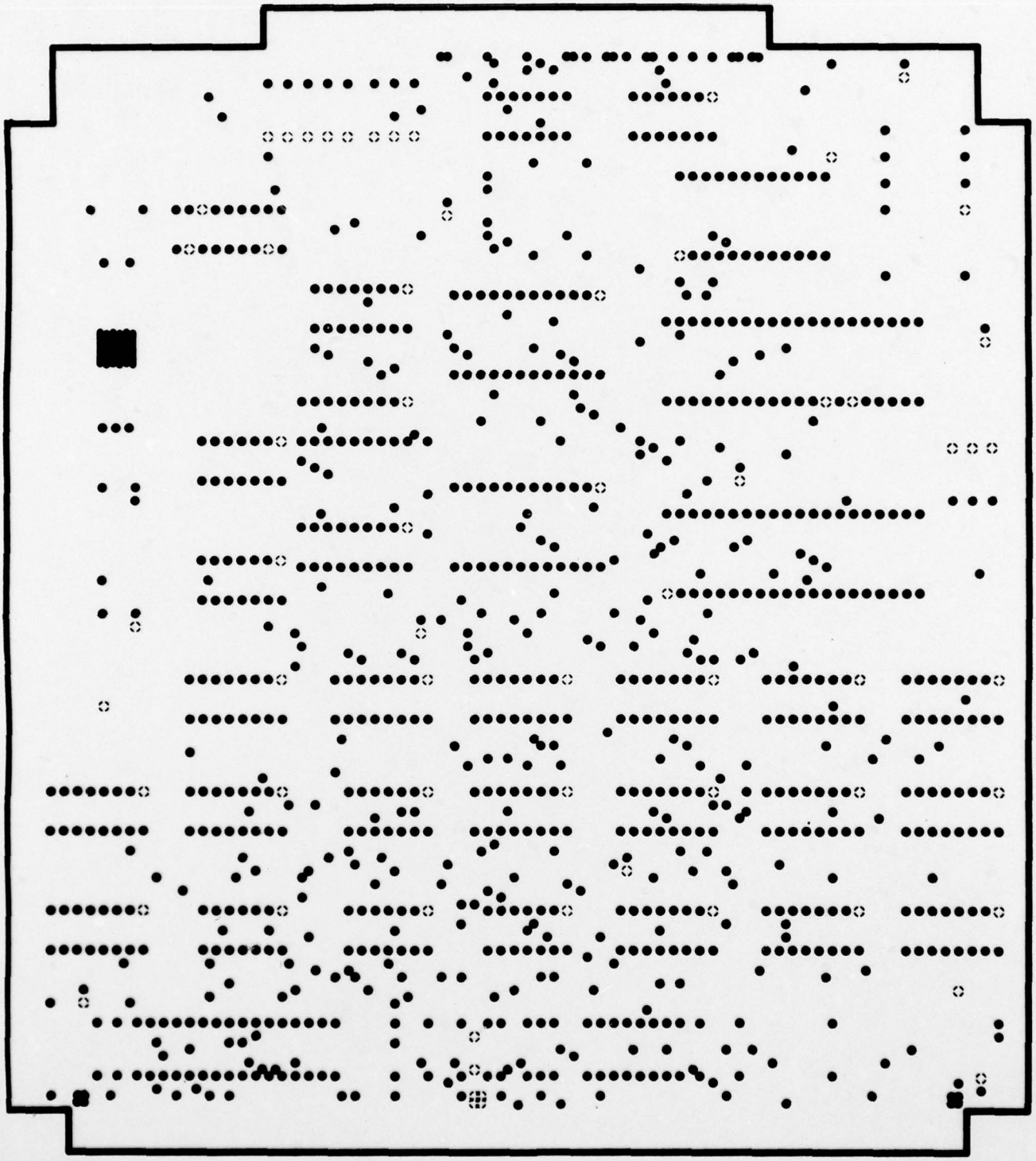

Figure 5(c): Sample Artwork, Power Plane

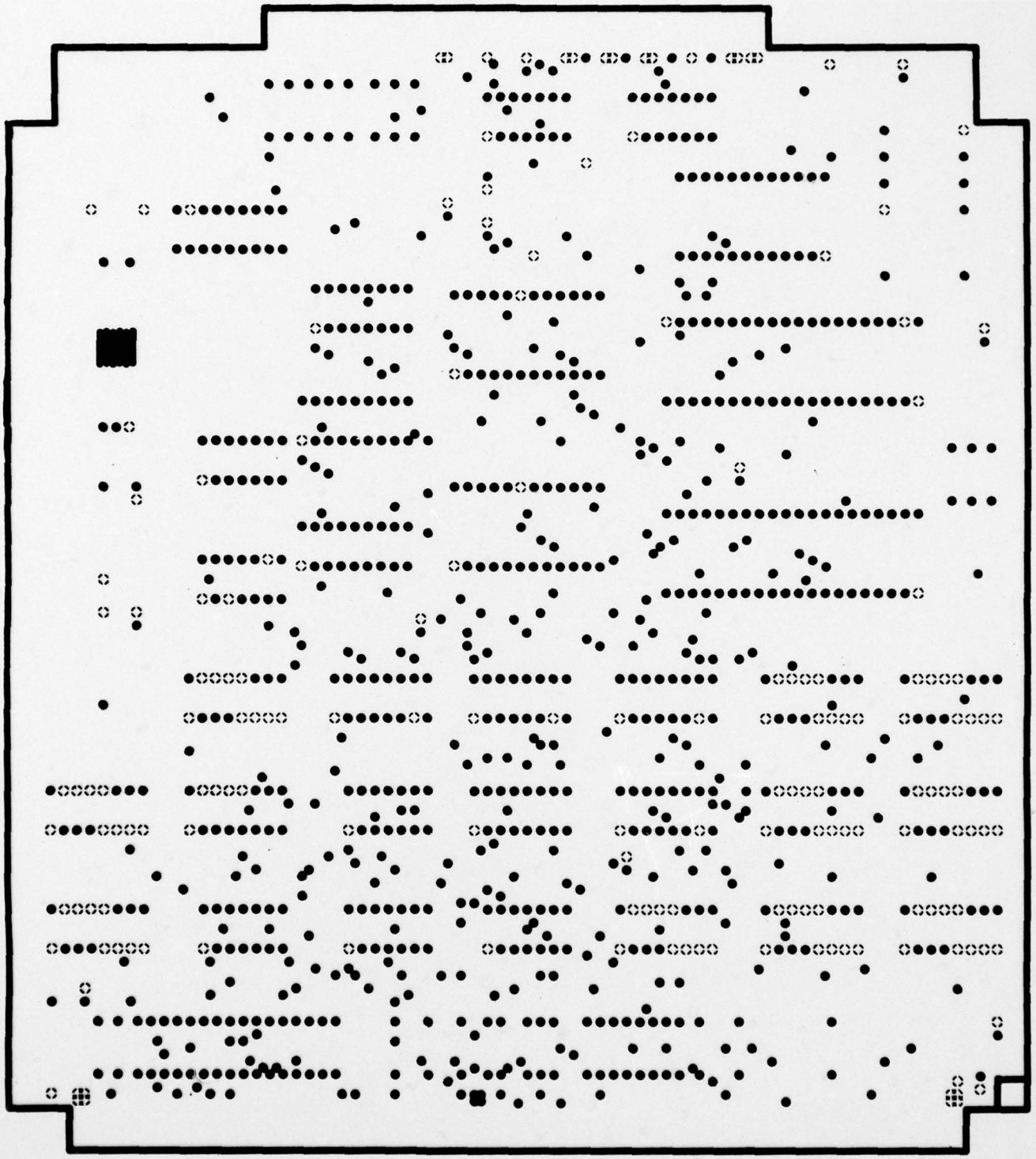

Figure 5(d): Sample Artwork, Ground Plane

Example #	# of IC's	# of discrete comps.	# of pins	# of nets	board size (in.)	Grid size (in.)	time for placer (min.)	CPU time placer (sec)	CPU time router (sec)	completion rate (%)	CPU time via elim. (sec)	via elim. rate (%)
1	49	28	808	210	9.3 7.6	.05	20	45	58	99	14	44
2	72	6	1286	282	7.0 9.0	.033	34	149	85	100	35	41
3	20	37	610	133	4.7 6.4	.05	31	41	36	100	14	52
4	12	4	200	63	4.0 3.0	.05	8	5	6	100	3	61
5	22	1	466	162	5.0 5.6	.05	10	16	84	96	9	43
6	27	101	729	185	9.5 5.5	.05	55	30	53	100	12	45

Table I. Summary of results

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Note No. 131	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SPRINT - An Interactive System for Printed Circuit Board Design: Functional Characteristics		5. TYPE OF REPORT & PERIOD COVERED Technical Note 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) W. M. vanCleemput, T. C. Bennett, J. A. Hupp, K. R. Stevens		8. CONTRACT OR GRANT NUMBER(s) N-00014-75-C-0601
9. PERFORMING ORGANIZATION NAME AND ADDRESS Stanford Electronics Laboratories Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of the Naval Research Department of the Navy Washington, DC 22217		12. REPORT DATE November 1977 13. NO. OF PAGES 27 15. SECURITY CLASS. (of this report)
14. MONITORING AGENCY NAME & ADDRESS (if diff. from Controlling Office)		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this report)

Reproduction in whole or part is permitted for any purpose of the United State Government.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer-aided design, printed circuit boards, structured design, design automation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The SPRINT system, designed and implemented on the IBM 370 system at the Stanford Linear Accelerator Center (SLAC), is a highly interactive system for the layout of printed circuit boards.

At the choice of the designer, a layout can be generated manually, automatically or usually by a combination of man-machine interaction and automatic design algorithms.

DD FORM 1 JAN 73 1473
EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

JSEP REPORTS DISTRIBUTION LIST

Department of Defense

Director
National Security Agency
Attn: Dr. T. J. Beahn
Fort George G. Meade
Maryland 20755

Defense Documentation Center (12)
Attn: DDC-TCA (Mrs. V. Caponio)
Cameron Station
Alexandria, Virginia 22314

Assistant Director
Electronics and Computer Sciences
Office of Director of Defense
Research and Engineering
The Pentagon
Washington, D.C. 20315

Defense Advanced Research
Projects Agency
Attn: Dr. R. Reynolds
1400 Wilson Boulevard
Arlington, Virginia 22209

Department of the Army

Commandant
US Army Air Defense School
Attn: ATSAD-T-CSM
Fort Bliss, Texas 79916

Commander
US Army Armament R&D Command
Attn: DRDAR-TSS
Dover, New Jersey 07801

Commander
US Army Armament R&D Command (BRL)
Attn: DRDAR-TSB-S
Aberdeen Proving Ground
Aberdeen, Maryland 21005

Commandant
US Army Command and General Staff
College
Attn: Acquisitions, Library Division
Fort Leavenworth, Kansas 66027

Commander
US Army Communication Command
Attn: CC-OPS-PD
Fort Huachuca, Arizona 85613

Commander
US Army Materials and Mechanics
Research Center
Attn: Chief, Materials Science Div.
Watertown, Massachusetts 02172

Commander
US Army Materiel Development and
Readiness Command
Attn: Technical Library, Rm. 7S 35
5001 Eisenhower Avenue
Alexandria, Virginia 22333

Commander
US Army Missile R&D Command
Attn: Chief, Document Section
Redstone Arsenal, Alabama 35809

Commander
US Army Satellite Communications Agency
Fort Monmouth, New Jersey 07703

Director
US Army Signals Warfare Laboratory
Attn: DELSW-OS
Arlington Hall Station
Arlington, Virginia 22212

Project Manager
ARTADS
EAI Building
West Long Branch, New Jersey 07764

NOTE: One (1) copy to each addressee unless otherwise indicated.

Commander/Director
Atmospheric Sciences Lab. (ECOM)
Attn: DRSEL-BL-DD
White Sands Missile Range
New Mexico 88002

Commander
US Army Electronics Command
Attn: DRSEL-NL-O (Dr. H. S. Bennett)
Fort Monmouth, New Jersey 07703

Director
TRI-TAC
Attn: TT-AD (Mrs. Brittler)
Fort Monmouth, New Jersey 07703

Commander
US Army Electronics Command
Attn: DRSEL-CT-L (Dr. R. Buser)
Fort Monmouth, New Jersey 07703

Director
Electronic Warfare Lab. (ECOM)
Attn: DRSEL-WL-MY
White Sands Missile Range
New Mexico 88002

Executive Secretary, TAC/JSEP
US Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

Commander
Harry Diamond Laboratories
Attn: Mr. John E. Rosenberg
2800 Powder Mill Road
Adelphi, Maryland 20783

HQDA (DAMA-ARZ-A)
Washington, D.C. 20310

Commander
US Army Electronics Command
Attn: DRSEL-TL-E (Dr. J. A. Kohn)
Fort Monmouth, New Jersey 07703

Commander
US Army Electronics Command
Attn: DRSEL-TL-EN
(Dr. S. Kroenenberg)
Fort Monmouth, New Jersey 07703

Commander
US Army Electronics Command
Attn: DRSEL-NL-T (Mr. R. Kulinyi)
Fort Monmouth, New Jersey 07703

Commander
US Army Electronics Command
Attn: DRSEL-NL-B (Dr. E. Lieblein)
Fort Monmouth, New Jersey 07703

Commander
US Army Electronics Command
Attn: DRSEL-TL-MM (Mr. N. Lipetz)
Fort Monmouth, New Jersey 07703

Commander
US Army Electronics Command
Attn: DRSEL-RD-O (Dr. W. S. McAfee)
Fort Monmouth, New Jersey 07703

Director
Night Vision Laboratory
Attn: DRSEL-NV-D
Fort Belvoir, Virginia 22060

Col. Robert Noce
Senior Standardization Representative
US Army Standardization Group, Canada
Canadian Force Headquarters
Ottawa, Ontario, CANADA KIA)K2

Commander
US Army Electronics Command
Attn: DRSEL-NL-B (Dr. D. C. Pearce)
Fort Monmouth, New Jersey 07703

Commander
US Army Electronics Command
Attn: DRSEL-NL-RH-1
(Dr. F. Schwering)
Fort Monmouth, New Jersey 07703

Commander
US Army Electronics Command
Attn: DRSEL-TL-I
(Dr. C. G. Thornton)
Fort Monmouth, New Jersey 07703

US Army Research Office (3)
Attn: Library
P. O. Box 12211
Research Triangle Park
North Carolina 27709

Director
Division of Neuropsychiatry
Walter Reed Army Institute
of Research
Washington, D.C. 20012

Commander
White Sands Missile Range
Attn: STEWS-ID-R
White Sands Missle Range
New Mexico 88002

Department of the Air Force

Mr. Robert Barrett
RADC/ETS
Hanscom AFB, Massachusetts 01731

Dr. Carl E. Baum
AFWL (ES)
Kirtland AFB, New Mexico 87117

Dr. E. Champagne
AFAL/DH
Wright-Patterson AFB, Ohio 45433

Dr. R. P. Dolan
RADC/ETSD
Hanscom AFB, Massachusetts 01731

Mr. W. Edwards
AFAL/TE
Wright-Patterson AFB, Ohio 45433

Professor R. E. Fontana
Head, Department of Electrical
Engineering
AFIT/ENE
Wright-Patterson AFB, Ohio 45433

Dr. Alan Garscadden
AFAPL/POD
Wright-Patterson AFB, Ohio 45433

USAF European Office of Aerospace
Research
Attn: Major J. Gorrell
Box 14, FPO, New York 09510

LTC Richard J. Gowen
Department of Electrical Engineering
USAF Academy, Colorado 80840

Mr. Murray Kesselman (ISCA)
Rome Air Development Center
Griffiss AFB, New York 13441

Dr. G. Knausenberger
Air Force Member, TAC
Air Force Office of Scientific
Research, (AFSC) AFSOR/NE
Bolling Air Force Base, DC 20332

Dr. L. Kravitz
Air Force Member, TAC
Air Force Office of Scientific
Research, (AFSC) AFSOR/NE
Bolling Air Force Base, DC 20332

Mr. R. D. Larson
AFAL/DHR
Wright-Patterson AFB, Ohio 45433

Dr. Richard B. Mack
RADC/ETER
Hanscom AFB, Massachusetts 01731

Mr. John Mottsmith (MCIT)
HQ ESD (AFSC)
Hanscom AFB, Massachusetts 01731

Dr. Richard Picard
RADC/ETSL
Hanscom AFB, Massachusetts 01731

Dr. J. Ryles
Chief Scientist
AFAL/CA
Wright-Patterson AFB, Ohio 45433

Dr. Allan Schell
RADC/ETE
Hanscom AFB, Massachusetts 01731

Mr. H. E. Webb, Jr. (ISCP)
Rome Air Development Center
Griffisss AFB, New York 13441

LTC G. Wepfer
Air Force Office of Scientific
Research, (AFSC) AFOSR/NP
Bolling Air Force Base, DC 20332

LTC G. McKemie
Air Force Office of Scientific
Research, (AFSC) AFOSR/NM
Bolling Air Force Base, DC 20332

Department of the Navy

Dr. R. S. Allgaier
Naval Surface Weapons Center
Code WR-303
White Oak
Silver Spring, Maryland 20910

Naval Weapons Center
Attn: Code 5515, H. F. Blazek
China Lake, California 93555

Dr. H. L. Blood
Technical Director
Naval Undersea Center
San Diego, California 95152

Naval Research Laboratory
Attn: Code 5200, A. Brodzinsky
4555 Overlook Avenue, SW
Washington, D.C. 20375

Naval Research Laboratory
Attn: Code 7701, J. D. Brown
4555 Overlook Avenue, SW
Washington, D.C. 20375

Naval Research Laboratory
Attn: Code 5210, J. E. Davey
4555 Overlook Avenue, SW
Washington, D.C. 20375

Naval Research Laboratory
Attn: Code 5460/5410, J. R. Davis
4555 Overlook Avenue, SW
Washington, D.C. 20375

Naval Ocean Systems Center
Attn: Code 75, W. J. Dejka
271 Catalina Boulevard
San Diego, California 92152

Naval Weapons Center
Attn: Code 601, F. C. Essig
China Lake, California 93555

Naval Research Laboratory
Attn: Code 5510, W. L. Faust
4555 Overlook Avenue, SW
Washington, D.C. 20375

Naval Research Laboratory
Attn: Code 2626, Mrs. D. Folen
4555 Overlook Avenue, SW
Washington, D.C. 20375

Dr. Robert R. Fossum
Dean of Research
Naval Postgraduate School
Monterey, California 93940

Dr. G. G. Gould
Technical Director
Naval Coastal System Laboratory
Panama City, Florida 32401

Naval Ocean Systems Center
Attn: Code 753, P. H. Johnson
271 Catalina Boulevard
San Diego, California 92152

Donald E. Kirk
Professor and Chairman
Electronic Engineer, SP-304
Naval Postgraduate School
Monterey, California 93940

Naval Air Development Center
Attn: Code 01, Dr. R. K. Lobb
Johnsville
Warminster, Pennsylvania 18974

Naval Research Laboratory
Attn: Code 5270, B. D. McCombe
4555 Overlook Avenue, SW
Washington, D.C. 20375

Capt. R. B. Meeks
Naval Sea Systems Command, NC #3
2531 Jefferson Davis Highway
Arlington, Virginia 20362

Dr. H. J. Mueller
Naval Air Systems Command
Code 310, JP #1
1411 Jefferson Davis Highway
Arlington, Virginia 20360

Dr. J. H. Mills, Jr.
Naval Surface Weapons Center
Electronics Systems Department
Code DF
Dahlgren, Virginia 22448

Naval Ocean Systems Center
Attn: Code 702, H. T. Mortimer
271 Catalina Boulevard
San Diego, California 92152

Naval Air Development Center
Attn: Technical Library
Johnsville
Warminster, Pennsylvania 18974

Naval Ocean Systems Center
Attn: Technical Library
271 Catalina Boulevard
San Diego, California 92152

Naval Research Laboratory
Underwater Sound Reference Division
Technical Library
P. O. Box 8337
Orlando, Florida 32806

Naval Surface Weapons Center
Attn: Technical Library
Code DX-21
Dahlgren, Virginia 22448

Naval Surface Weapons Center
Attn: Technical Library
Building 1-330, Code WX-40
White Oak Laboratory
Silver Spring, Maryland 20910

Naval Training Equipment Center
Attn: Technical Library
Orlando, Florida 32813

Naval Undersea Center
Attn: Technical Library
San Diego, California 92152

Naval Underwater Systems Center
Attn: Technical Library
Newport, Rhode Island 02840

Office of Naval Research
Electronic and Solid State
Sciences Program (Code 427)
800 North Quincy Street
Arlington, Virginia 22217

Office of Naval Research
Mathematics Program (Code 432)
800 North Quincy Street
Arlington, Virginia 22217

Office of Naval Research
Naval Systems Division
Code 220/221
800 North Quincy Street
Arlington, Virginia 22217

Director
Office of Naval Research
New York Area Office
715 Broadway, 5th Floor
New York, New York 10003

Office of Naval Research
San Francisco Area Office
One Hallidie Plaza, Suite 601
San Francisco, California 94102

Director
Office of Naval Research Branch Office
495 Summer Street
Boston, Massachusetts 02210

Director
Office of Naval Research Branch Office
536 South Clark Street
Chicago, Illinois 60605

Director
Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91101

Mr. H. R. Riedl
Naval Surface Weapons Center
Code WR-34
White Oak Laboratory
Silver Spring, Maryland 20910

Naval Air Development Center
Attn: Code 202, T. J. Shopple
Johnsville
Warminster, Pennsylvania 18974

Naval Research Laboratory
Attn: Code 5403, J. E. Shore
4555 Overlook Avenue, SW
Washington, D.C. 20375

A. L. Slafkovsky
Scientific Advisor
Headquarters Marine Corps
MC-RD-1, Arlington Annex
Washington, D.C. 20380

Harris B. Stone
Office of Research, Development,
Test and Evaluation
NOP-987
The Pentagon, Room 5D760
Washington, D.C. 20350

Mr. L. Sumney
Naval Electronics Systems Command
Code 3042, NC #1
2511 Jefferson Davis Highway
Arlington, Virginia 20360

David W. Taylor
Naval Ship Research and
Development Center
Code 522.1
Bethesda, Maryland 20084

Naval Research Laboratory
Attn: Code 4105, Dr. S. Teitler
4555 Overlook Avenue, SW
Washington, D.C. 20375

Lt. Cdr. John Turner
NAVMAT 0343
CP #5, Room 1044
2211 Jefferson Davis Highway
Arlington, Virginia 20360

Naval Ocean Systems Center
Attn: Code 746, H. H. Wieder
271 Catalina Boulevard
San Diego, California 92152

Dr. W. A. Von Winkle
Associate Technical Director for
Technology
Naval Underwater Systems Center
New London, Connecticut 06320

Dr. Gernot M. R. Winkler
Director, Time Service
US Naval Observatory
Massachusetts Ave. at 34th St., NW
Washington, D.C. 20390

Other Government Agencies

Dr. Howard W. Etzel
Deputy Director
Division of Materials Research
National Science Foundation
1800 G Street
Washington, D.C. 20550

Mr. J. C. French
National Bureau of Standards
Electronics Technology Division
Washington, D.C. 20234

Dr. Jay Harris
Program Director
Devices and Waves Program
National Science Foundation
1800 G Street
Washington, D.C. 20550

Los Alamos Scientific Laboratory
Attn: Reports Library
P. O. Box 1663
Los Alamos, New Mexico 87544

Dr. Dean Mitchell
Program Director
Solid-State Physics
Division of Materials Research
National Science Foundation
1800 G Street
Washington, D.C. 20550

Mr. F. C. Schwenk, RD-T
National Aeronautics and Space
Administration
Washington, D.C. 20546

M. Zane Thornton
Deputy Director, Institute for
Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

Director
Stanford Electronics Laboratories
Stanford University
Stanford, California 94305

Nongovernment Agencies

Director
Columbia Radiation Laboratory
Columbia University
538 West 120th Street
New York, New York 10027

Director
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

Director of Laboratories
Division of Engineering and
Applied Physics
Harvard University
Pierce Hall
Cambridge, Massachusetts 02138

Director
Electronics Research Center
The University of Texas
Engineering-Science Bldg. 112
Austin, Texas 78712

Director
Electronics Research Laboratory
University of California
Berkeley, California 94720

Director
Electronics Sciences Laboratory
University of Southern California
Los Angeles, California 90007

Director
Microwave Research Institute
Polytechnic Institute of New York
333 Jay Street
Brooklyn, New York 11201

Director
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Officer in Charge
Carderock Laboratory
Code 18, G. H. Gleissner
David Taylor Naval Ship Research
and Development Center
Bethesda, Maryland 20084

Dr. Roy F. Potter
3868 Talbot Street
San Diego, California 92106